Dear Federal Communications Commissioners,
I write in support of Network Neutrality. This comment includes a copy, appended below my signature, of my comment to the Federal Trade Commission on the same issue, after the FTC's 13-14 February 2007 FTC Workshop on Network Neutrality. Before considering either legislation or regulatory action, the FCC should understand clearly what the Net is, and how the Net differs from any bundle of services provided by any ISP. The Net is larger and more important, and the issue of Net Neutrality is an issue of fundamental rights, rights of privacy, free association, free speech, and free enterprise. My comment explains what the Net is, and what the Net is not.

My comment is long because it incorporates the Internet Assigned Numbers Authority list of ports. My actual comment is about twenty pages long.

I remain your fellow computer owner, fellow user of the Net, and fellow citizen of the United States of America,
Jay Sulzberger.

Comments on the definition of the Internet and the importance of distinguishing the Net from cable TV, and from lower level signal transport systems, after the FTC Workshop of 13 and 14 February 2007 on Network Neutrality

I am Jay Sulzberger. I am a working member of New Yorkers for Fair Use, and I attended the 13-14 February 2007 FTC Workshop on Network Neutrality. This comment is not an official statement of New Yorkers for Fair Use.

The full name of the FTC workshop on 13 and 14 February 2007 included the phrase "broadband competition". At the workshop, most of the discussions failed to address the real issue of Network Neutrality because no correct definition of the Net was presented.

Internet packets may be carried over slow communications links or over

fast links. Most week days I use several different connections to the Net. Some days I use a dialup connection, which is much slower than the connection over lines owned by the Cable Company, or by the City of New York. But no matter whether I use slow lines or fast lines, I use the Net for private, for tribal, for business, and for public communications. And I use the Net in freedom. And some days I watch cable TV.

Now cable TV is not the Internet, but most speakers at the FTC's workshop spoke of the Net in ways that treated it as if it were just a form of interactive TV, with some extra special services bundled with interactive TV, "web viewing", email, and doubtfully, voice over IP.

Questions around building another, perhaps several other, cable TV networks, are not part of the issue of Network Neutrality, because the Net is not TV of any kind.

Use of the word "broadband" to mean both the Net and cable TV helps perpetuate the fundamental confusion.

We now state explicitly the main implicit error which underlies the FTC's present failure to engage with the issue of Network Neutrality:

Main Implicit Error: The Internet is not some bundle of services delivered by the Telephone Company and/or the Cable Company.

The Net is difficult to understand unless you work with it productively to develop applications and therefore already know what it is; and in that case, it is still difficult to convey what the Net is, because some history and background information, is needed to grasp how different the Net is from cable TV.

I admit that many people who today visit web sites and do email do not clearly understand what the Net is. But certainly the FTC should clearly understand what the Net is.

When ranting on the topic of the Net in years past, I have often

claimed that one can explain the Net without knowing anything of matters denominated in the popular press as "technical". After the workshop, I now think that complete ignorance can impede understanding of the Net. One speaker at the workshop who spoke against Net Neutrality had never heard of a port. It is likely that part of his, and others, opposition to Net Neutrality is due to simple ignorance.

We distinguish several faces of the Net.

First let us define what the Net is for a user of the Net. This face of the Net is simple to explain. But the explanation may, at first, be hard to understand, not because of subtleties, but rather just the opposite, because of the simplicity and the extremity of the defining principles.

If you do not know what a port is, then likely you implicitly assume that the Duopoly invented the Net and gave us the World Wide Web, email, videos over the Net, massive virtual worlds/games, etc.. So, what is a port? For a good introduction to ports, see

http://en.wikipedia.org/wiki/TCP_and_UDP_port
http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

Consider me sitting in front of my computer. My computer is connected to the Net. My computer has an Internet address, say 192.0.34.166. Every computer connected to the Net has such an address. (For now, the question of how long the hardware keeps the same address, and who decides the address, how other computers figure out the address, etc. are all irrelevant.)

Important point: We assume in this explanation that I am in full control of my computer.

Now I issue, as root, this command to my computer

nmap -sT -O localhost

and I get back:

```
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2007-02-26 19:25 EST
Interesting ports on localhost.localdomain (127.0.0.1):
Not shown: 1656 closed ports
PORT      STATE SERVICE
1/tcp     open  tcpmux
11/tcp    open  systat
15/tcp    open  netstat
25/tcp    open  smtp
79/tcp    open  finger
111/tcp   open  rpcbind
119/tcp   open  nntp
143/tcp   open  imap
540/tcp   open  uucp
635/tcp   open  unknown
993/tcp   open  imaps
1080/tcp  open  socks
1524/tcp  open  ingreslock
2000/tcp  open  callbook
6667/tcp  open  irc
12345/tcp open  NetBus
12346/tcp open  NetBus
27665/tcp open  Trinoo_Master
31337/tcp open  Elite
32771/tcp open  sometimes-rpc5
32772/tcp open  sometimes-rpc7
32773/tcp open  sometimes-rpc9
32774/tcp open  sometimes-rpc11
54320/tcp open  bo2k
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.5.25 - 2.6.8 or Gentoo 1.2 Linux 2.4.19 rc1-rc7, Linux 2.6.3 - 2.6.10
Uptime 8.412 days (since Sun Feb 18 09:32:55 2007)

Nmap finished: 1 IP address (1 host up) scanned in 2.184 seconds
```

Ports, in the jargon of Internet engineers, are entrances and/or exits of my computer. They are entrances/exits for data. nmap is a program which knocks on every port of the specified computer. The knocks are delivered over the Net, that is, nmap is an Internet application. I specified in my command that nmap knock on the ports of my computer, here called "localhost". (Irrelevant detail: from the vantage of my computer itself, my computer has by convention, sometimes, the Internet address 127.0.0.1.) nmap reports back which ports are open to incoming connections, that is, which ports will accept data coming in. nmap may be used to knock on the ports of any computer on the Net, even though in this example, I instructed nmap to only check my own computer.

The three columns of the first part of the data returned by nmap are:

1. the port, given by a number and what sort of data packets the port sends/receives

2. the state of the port

3. what service the people who built and run the Net have agreed should usually be listening at the port

The next part of the information returned by nmap, that is,

```
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.5.25 - 2.6.8 or Gentoo 1.2 Linux 2.4.19 rc1-rc7, Linux 2.6.3 - 2.6.10
Uptime 8.412 days (since Sun Feb 18 09:32:55 2007)
```

is nmap's guess as to what sort of computer system nmap knocked on. The guess is that my computer is a general purpose computer running some version of the GNU/Linux operating system, and that it has been up "8.412 days". (The uptime is not correct. nmap is not infallible.)

Finally nmap tells what it did:

```
Nmap finished: 1 IP address (1 host up) scanned in 2.184 seconds
```

The information returned by nmap helps to show us what the Net is. My machine is connected to the Net. My machine has an address which allows other machines on the Net to send data to my machine in order, perhaps, to start two streams of data flowing between the two machines. Suppose that you also have before you a computer connected to the Net. Again, we assume that you have full control of your box.

We now present a well known, popular, and paradigmatic, use of the Internet:

You might have opened port 80 on your box, and placed a program called an "http daemon", say Apache, to listen for incoming requests on port 80. I might start a Web browser, say Firefox, on my machine and tell my browser to ask for a web page on your machine. Firefox would now knock on port 80 on your machine, and request the page.

This is the mechanism by which web pages are delivered. Between your machine and mine lies the Net. Today, within the Net, there is no third party wiretapping our communication. There is no third party attempting to determine the value of our conversation to us, in order to charge us extra for valuable conversations. There is no third party examining our conversation in order to silence us if we say things the wiretapper's boss finds unpleasant.

I should have said "In the best circumstances, there is no ..." because, if your machine were in the United States, and mine in China, and if our conversation were to include mention of movements not favored by the government of China, then China might act to stop our conversation. And if your machine were in the United States and your "Internet Service Provider" refused to pass that first packet to port 80 on your machine, then our conversation could not even get started. The policy of refusing that first packet to port 80 is often called the "no servers allowed" rule. Both interferences result in something that is not the full and true Internet.

Let us list two other uses of the Net:

1. You might want to send me some email. By convention, usually port 25 is used for email. If I choose to get email by the usual means, I must open port 25, and place a "mail transport agent", say exim, listening on the port. But we both might want to send email back and forth by other means. In that case, you and I might have agreed before hand to use another port, or set of ports, and we might have agreed on some other protocol for delivery of email. We are free to make our own arrangements, without asking permission of any third party.

2. Assume you are a master sysadmin and an old friend. I might be having trouble with my computer, and I might ask you, by telephone say, to come into my box and have a look around in order to help me. We might have agreed that you would come in via "secure shell", that is, ssh. By convention, the usual port for ssh is port 22, and when I have opened up port 22, and set the ssh daemon listening, and granted you certain privileges on my system, you may run ssh on your box, ssh will open a connection to my box and you may now log in to my box. But we might, for some reason, choose not to use ssh and port 22. In that case, you and I might have agreed before hand to use another port, or set of ports, and we might have agreed on some other protocol by which you can log into my machine. We are free to make our own arrangements, without asking permission of any third party.

Here is the definition of the Internet. The definition comes in three parts.

1. My machine and yours stand equal as peers, in the sense that my or your machine can initiate, or attempt to initiate a conversation, without interference by third parties. And my machine or yours can accept such an invitation to converse, without interference by third parties. Naturally if your machine attempts to make a connection and my machine refuses, this is still the Net. But, up to bugs, communication occurs, or fails, by agreement, or disagreement, between you and me, as long as we have paid our ISPs.

2. If you and I agree on a protocol, and agree on a port, or ports, for our data streams to travel over, and we write, or otherwise

obtain, two programs that use the protocol, we may set our programs running on our boxes, and we may communicate using the protocol we have together freely chosen, without interference by third parties. In particular, we are free to invent new protocols, and to write programs using the new protocols, and to set these programs running over the Net.

3. About one billion machines are connected and use the freely agreed on protocols that enable our conversations to be transported by the wires, fibers, through the routers, over the air, via satellites, carrier pigeons. And it all works because many people and companies and governments have built the pieces and connected them so that it does work to make possible the extraordinarily flexible and free communication specified in 1 and 2. The main underlying protocol of the Net is TCP/IP. Note that there are other networks which use the TCP/IP protocol, but are not the Net, because they do not allow on the order of one billion people to freely connect with each other with complete flexibility in their manner of communication, that is, in the protocols running atop TCP/IP. For example, various variants of cable TV may be delivered using the TCP/IP protocol. But such a cable TV network is not part of the Net.

Now we come face to face with the Main Error. If the Net were some bundle of services provided by the Duopoly, and perhaps, in future, by some other companies, then none of these defining conditions of the Net could be met. In order to be brief, we shall only explain why condition 2 would fail, if the Duopoly ran the Net:

ad 2. If the Duopoly provided the Net, then surely you and I could not design and deploy and use a new protocol without either working for the Duopoly, or negotiating with the Duopoly some arrangement by which the Duopoly would support the protocol. But this is exactly not how the Net was built. We adduce here a wonderful document, called the "port numbers", which may be found at the website of the IANA, the Internet Assigned Numbers Authority:

http://www.iana.org/assignments/port-numbers

We append one version of this list, it is on occasion updated, as Document IANA-port-numbers below.

Let us look at some lines from the list of ports:

```
  Keyword        Decimal    Description                  References
  -------        -------    -----------                  ----------
                 0/tcp    Reserved
                 0/udp    Reserved
  #                      Jon Postel <postel@isi.edu>
  tcpmux           1/tcp    TCP Port Service Multiplexer
  tcpmux           1/udp    TCP Port Service Multiplexer
  #                      Mark Lottor <MKL@nisc.sri.com>
  compressnet      2/tcp    Management Utility
  compressnet      2/udp    Management Utility
  compressnet      3/tcp    Compression Process
  compressnet      3/udp    Compression Process
  #                      Bernie Volz <volz@cisco.com>
  #                4/tcp    Unassigned
  #                4/udp    Unassigned
  rje              5/tcp    Remote Job Entry
  rje              5/udp    Remote Job Entry
  #                      Jon Postel <postel@isi.edu>
  #                6/tcp    Unassigned
  #                6/udp    Unassigned
  echo               7/tcp    Echo
  echo               7/udp    Echo
  #                      Jon Postel <postel@isi.edu>
  #                8/tcp    Unassigned
  #                8/udp    Unassigned
  discard          9/tcp    Discard
  discard          9/udp    Discard
  #                      Jon Postel <postel@isi.edu>
  discard          9/dccp   Discard SC:DISC
  #                      IETF dccp WG, Eddie Kohler <kohler@cs.ucla.edu>, [RFC4340]
  #               10/tcp    Unassigned
  #               10/udp    Unassigned
  systat           11/tcp    Active Users
```

```
systat          11/udp    Active Users
#                       Jon Postel <postel@isi.edu>
#               12/tcp    Unassigned
#               12/udp    Unassigned
daytime         13/tcp    Daytime (RFC 867)
daytime         13/udp    Daytime (RFC 867)
#                       Jon Postel <postel@is
```